(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 248 403 B1**

(12) # EUROPEAN PATENT SPECIFICATION

(45) Date of publication of patent specification: **13.04.94**

(51) Int. Cl.[5]: **G06F 9/46**, G06F 15/16

(21) Application number: **87107964.6**

(22) Date of filing: **02.06.87**

(54) **Distributed interactive processing method in complex system including plural work stations and plural host computers and apparatus using the same.**

(30) Priority: **02.06.86 JP 125726/86**

(43) Date of publication of application:
09.12.87 Bulletin 87/50

(45) Publication of the grant of the patent:
13.04.94 Bulletin 94/15

(84) Designated Contracting States:
DE FR GB

(56) References cited:
EP-A- 0 108 899

PROCEEDINGS 1st INTERNATIONAL CONFERENCE ON COMPUTER WORKSTATIONS, San Jose, California, 11th-14th November 1985, pages 198-208, IEEE, New York, US; R. AGRAWAL et al.: "Processor sharing in nest: A network of computer workstations"

PROCEEDINGS OF THE 1985 INTERNATIONAL CONFFERENCE ON PARALLEL PROCESSING, 20th-23rd August 1985, pages 139-142, IEEE, New York, US; A.K. EZZAT et al.: "Making oneself known in a distributed world"

(73) Proprietor: **HITACHI, LTD.**
6, Kanda Surugadai 4-chome
Chiyoda-ku, Tokyo 101(JP)

(72) Inventor: **Masai, Kazuo**
31-14, Hino-7-chome
Konan-ku Yokohama(JP)
Inventor: **Yamada, Kimitoshi**
4-14, Soya-2-chome
Hadano-shi(JP)
Inventor: **Kojima, Tomihiko**
592-4, Miwamachi
Machida-shi(JP)
Inventor: **Kimura, Ikuo**
5-18, Minamirinkan-3-chome
Yamato-shi(JP)

(74) Representative: **Patentanwälte Beetz - Timpe - Siegfried Schmitt-Fumian - Mayr**
Steinsdorfstrasse 10
D-80538 München (DE)

EP 0 248 403 B1

IBM TDB, vol. 23, no. 5, October 1980, page 1811-1812, J.G. Sams: "Node processor for distributed system control".

B.W. Kernighan et al.: "The UNIX programming environment", Prentice-Hall, Inc. 1984.

Description

BACKGROUND OF THE INVENTION

The present invention relates to complex systems including intelligent work stations (shortly referred to as IWS's herebelow) and host computers (shortly referred to as hosts herebelow), and it is particularly concerned with a distributed interactive processing method suitable to distribute the load on the hosts to the IWS's in such a system as well as with the means for performing the said method.

In an interactive processing system such as the time sharing system (TSS), for the user's request issued from a terminal equipment, the syntactical check has been effected on the terminal side as described in the JP-A-60-144839. However, the request itself is analyzed in the virtual address space of a host computer to select a program satisfying the request in the same space. The selected program first arranges the necessary environment, for example, allocates file and effects an OPEN processing thereon, and then executes the requested processing.

PROCEEDINGS 1st INTERNATIONAL CONFERENCE ON COMPUTER WORKSTATIONS, San Jose, California, 11th-14th November 1985, pages 198-208, IEEE, New York report on a project for a network of highly autonomous yet cooperating personal computer work stations and shared servers. An important aspect of the said project is to provide processor sharing by creating a pool of compute servers in the network that may be used by the work stations to supplement their computing needs. Some processors are permanently designated to be the compute servers, and in addition, through an advertisement mechanism, any work station may make itself temporarily available for a specific duration of time to be used as a compute server. The main object of the said project is to supplement the computing capability.

The said project is also dealt with in PROCEEDINGS OF THE 1985 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSINGm 20-23rd August 1985, pages 139-142, IEEE, New York. There it is indicated that the system is designed to allow processors to advertise themselves as available to some or all nodes in the network and withdraw as a compute server in a fully distributed manner.

The gist of the said project demands that the computing environment is decentralised and the network resources are dynamically configured. The work stations are allowed to use the compute servers freely and they may make themselves available to be used as compute servers. Respective work stations designate respective compute servers they will use. The work stations use the compute servers by selecting them from the pool of available compute servers. The computing needs are shared among respective work stations and their own processings are executed by the use of free compute servers. Accordingly, respective work stations transmit the programs to the processors by themselves.

Further, IBM TECHNICAL DISCLOSURE BULLETIN, vol. 23, no. 5, October 1980, pages 1811-1812 describe a node processor for distributed system control which is designed to be used in a distributed data processing system for connecting a local processor to a communications network to which are connected a number of other independent local processors. The distributed processing system is of a type wherein the different processes or task sets that can be performed by the different local processors are assigned process names and communications are carried on within and between the different local processors by means of messages which are addressed to the different processes by their process names. The node processor accomplishes process name recognition and provides message acceptance and routing functions. A process table is provided on the side of the servers.

Generally, in a transaction processing system (on-line system) a request is issued from a terminal in a small unit called a transaction to a host computer. A transaction executes a processing in accordance with the request in a fixed fashion in a space where the pertinent environment is arranged in advance.

In the transaction processing system, the processing is effected in the unit of transaction, namely, in an intermittent manner and in a fixed fashion. Consequently, this system is suitable for use with an automatic banking machine and the like of a bank; however, the transaction processing system is not suitable for an irregular job including various kinds of unfixed data.

In other words, as shown in Fig. 8, regular job requests supplied from a great number of terminals, for example, several thousands in number are processed in a fixed fashion. The host is provided with a data communication program to be effected by a terminal control section and job programs each for achieving a regular or fixed job and an input from a terminal is processed by a job program structured for a fixed purpose.

In the TSS, as shown in Fig. 9, request indications inputted from a relatively small number of terminals i.e. about 100 in number are received in a logical space disposed for each terminal and is subjected to a request analysis, and then a processing program is loaded for execution. In the host, there are provided a terminal control program, a request analysis program to analyze requests

from the respective terminals, and processing programs corresponding to the respective requests. Since the request analysis program and the processing programs are prepared for each terminal, the TSS is suitable for an irregular job handling various kinds of data.

In the TSS and the transaction processing system, a terminal is assigned to a host computer and cannot be easily allocated to be commonly or randomly shared among a plurality of host systems.

In the TSS of the conventional technology, although the syntactical check may be considered to be achieved in a terminal equipment, the request analysis cannot be implemented therein; consequently, there is provided a process (virtual address space) for each terminal to execute programs therein, namely, the request from a terminal is analyzed in the space and the processing environment is established in the same space, thereby executing a processing associated with the request. As a result, although the TSS is suitable for the irregular job to process various kinds of unfixed data, a large load is imposed on the host, which therefore cannot simultaneously process the requests from many terminals.

Furthermore, in the transaction processing system, there does not exist a connection between a terminal and a particular process (space) of the host computer, namely, there is not allocated a processing space dedicated to each terminal, and a routine or regular processing is effected in an intermittent fashion. Consequently, the load on the host can be reduced; however, irregular requests cannot be freely processed in the transastion processing system.

Moreover, in a system including a plurality of host computers, a terminal is required to be connected to a particular host before starting an interactive processing, which leads to a problem that terminals cannot concurrently and arbitrarily access a plurality of hosts for the services thereof.

SUMMARY OF THE INVENTION

The said object is achieved by means of a method as defined in the appended claim 1; advantageous further developments of the said method and apparatus for performing it respectively are defined in subclaims.

A user's request issued in an intelligent terminal is analyzed in the work station. When the work station recognizes that a particular server process in a particular host is necessary, a connection to the particular objective server process is established by means of a communication between processes. The server process sequentially receives processing requests from the process of the work station thus connected. Since the server process is a process prepared to realize or implement a predetermined kind of request, the processing can be effected without re-establishing the processing environment for each request. Thus, since one server process can receive processing requests from a plurality of terminals the host can assign an address space smaller than the number of the terminals thereby to reduce the address space of hosts.

When the kind of request from the user is changed, a new connection to another server process is established via the IWS and the processing is continued. In this situation, since the communication between processes (program executions) can be achieved with respect to any host, the server process may exist in an arbitrary host, and hence, in the case of a system including a plurality of hosts, the IWS can sequentially issue a request to the server process of each host to interactively perform an operation from the IWS.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

Fig. 1 is an overall system configuration diagram of a complex computer system including intelligent terminals and host computers according to the present invention;

Fig. 2 is a schematic block diagram illustrating the operation on the intelligent work station (IWS) side;

Fig. 3 is a schematic block diagram illustrating the operation on the host side;

Figs. 4 to 6 are flow charts respectively for explaining the operations of the screen control section, the request analysis/status control section, and process section of Fig. 2;

Fig. 7 is a block diagram exemplarily illustrating the configuration of the host;

Figs. 8 and 9 are explanatory diagrams for explaining the prior art TSS and on-line system, respectively; and

Fig. 10 is a conceptual diagram of the present invention useful for the comparison with the systems of Figs. 8 and 9.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A description will be given of an embodiment of the present invention with reference to the accompanying drawings.

In Fig. 1, there is shown an overall configuration of a distributed interactive processing system which comprises an intelligent work station side 1 and a host computer side 2 connected to each

other according to the present invention. Intelligent work stations (IWS's) 1_1, 1_2, ..., 1_n in the side 1 are connected to host computers 2_1, 2_2, ..., 2_m in the side 2 via communication lines 3_i (i = 1, 2, ..., ℓ). Each IWS 1_i includes a plurality of processes 4_i each for processing the user's request and a communication control section 5 for controlling communication lines 3_i. Each host computer comprises a host supervisor (program) providing a communication control space 6 for controlling the communication lines 3_1, 3_2, ... and a plurality of servers 7_1, 7_2, ..., 7_ℓ for executing processing associated with the processing requests. The servers 7_1, 7_2, ... are classified in accordance with the kinds associated with the processing requests from the process 4_i in the IWS 1_i, and a particular request from the process 4_i is executed by a particular server 7_i corresponding thereto.

Next, referring to Fig. 2 and subsequent diagrams, the operation of the present invention will be described.

Fig. 2 is a block diagram schematically illustrating processing blocks in the IWS 1_1 by way of example for IWS. The IWS 1_1 is connected to a display 8 and a mouse 9 as an input device for specifying a position on the display 8. The end user at the IWS 1_1 selects by means of the mouse 9 a graphic image called "Icon" displayed on the display device 8 in order to issue a request. The Icon is displayed as a graphic image associated with a name assigned to a resource such as a file, a program, or the like in a meaningful manner, thereby helping the user easily identify the resource. For example, a graphic image or picture illustrating a combination of a report sheet and a table is used as an Icon to represent a file storing data in a table format. The user identifies Icons representing objective files and programs for a desired operation among the Icons displayed on the screen of the IWS 1_1 and operates the mouse 7 to move the arrow-mark cursor to the pertinent Icon on the screen, thereby effecting an operation to select the Icon.

When the user selects one of Icons displayed on the screen by the screen control section 10 (step 41 of Fig. 4), the position select signal is inputted from the mouse 9 to the screen control section 10 of the IWS 1_1, and then the Icon number is obtained through the steps 42-47 of Fig. 4. The request analyze/status control section 11 receives the Icon number and the operation request. As shown in the steps 51-53 of Fig. 5, the request analyze/status control section 11 checks to determine a location where a processing for the selected Icon is to be effected or can be achieved, namely, based on the Icon number, an Icon attribute table 12 stored and developed in the storage of the IWS 1_1 in advance is searched for a process name corresponding to the relevant Icon. To the Icon attribute table 12, there have been registered entries each including a pair of a kind of Icon 12a and a name of process 12b to implement processings for the Icon. Consequently, the name of the process to execute the processing can be determined by effecting a retrieval through the Icon attribute table 12. The request analyze/status control section 11 initiates a process 4_i having the name thus determined (step 53).

The initiated process 4_i obtains names of the processing objectives such as a file name and a program name extracted from the Icon attribute table 12. Since the processing objectives are entirely registered to a directory 13 in the IWS 1_1 in advance, the locations thereof are determined by conducting a retrieval through the directory 13. The name of a processing objective comprises a name in which node names of the directory 13 are linked to each other, and hence the nodes of the relevant directory 13 can be directly traced by use of the name of the processing objective. As described above, the directory 13 is configured in a tree structure. Namely, in an upper node 13a of a node 13b of the determined objective, there is stored an existence location (indicating an existence processor name of a processor where the node 13b exists, or a destination processor name (host name) for other than the own processor). Consequently, the process 4_i can determine the destination or mate processor based on the name of the processing objective. The operation flow of the process 4_i is shown in the steps 61-67 of Fig. 6. Here, if the existence locations of the processing objectives are in the own IWS 1_1, the processing request can be accomplished in the IWS 1_1. If the existence locations are in another processor i.e. host, the directory 13 is searched to determine the name of the server 7_i in a host corresponding to the process 4_i in the IWS 1_1. It is here assumed that the name of the server 7_i is stored in the storage within the IWS 1_1 corresponding to the processing requested by the process 4_i. When the host name and the name of the server 7_i in the host are determined, the process 4_i issues a connection request to the communication control section 5 to establish a connection of IWS 1_1 to the objective server process 7_i. Based on the previously defined information such as a protocol, the communication control section 5 connects the communication line 3_1 to the destination processor i.e. relevant host computer 2_i. After the connection is completed, the communication control section 5 transfers a connection request buffer 14 to the host 2.

In order to establish a logical communication path between the process 4_i and the server process 7_i to be connected in the host 2_i, the commu-

nication control section 5 of the IWS 1_1 generates a connection request buffer 14 providing information indicative of a logical request for connecting the name of the process 4_i and the name of the destination server process 7_i, into a virtual memory of IWS 1_1. When a logical communication path is established as a result, the process 4_i in the IWS 1_1 can directly achieve communication with the server process 7_i of the host 2_i. The contents of request buffer area 14 are transferred to the communication control space 6 including a local supervisor program of the host 2_i via the communication line 3_1 connecting the IWS 1_1 to the host 2_i. The communication control space 6 has a function to accept a connection request and a processing request sent from the respective IWS 1_1 and to distribute the received requests to the corresponding ones of the server processes 7_1, 7_2, 7_3, ...

Referring now to Fig. 3, the configuration and operation of the host 2_1 will be described, by way of example.

The host 2_1 includes the communication control space 6 for receiving the connection and processing requests from the IWS 1_1 and for transferring the received requests to a relevant server 7_i.

Information received via the communication line 3_1 is stored in a buffer (area) existing in a virtual storage of the communication control space 6. The information sent from the IWS 1_1 includes the name of the process 4_i as the source and the name of the server process 7_i as the destination. In addition, information indicating the content of the request is also included in the sent information. When the content of the request is to establish a logical communication path, the communication control space 6 obtains the name of the server process 7_i in the host for which the communication path is to be established by use of the information received via the communication line 3_1. The server process 7_i is a process to implement a function on a particular operation objective contained in the request issued from the IWS 1_1 to the host 2_1. For example, the mail server accomplishes a function in the case where the operation objective is an electronic mail, namely, conducts the retrieval, distribution, and reception of the electronic mail. Moreover, there are provided the multi-media document server to store, to retrieve, and to extract the multi-media document; the print server to edit and to print the multi-media; and the data base server to retrieve, to store, and to extract the data base. At the initiation in the host 2, the server process 7_i conducts an initial report to the communication control space 6. Upon receiving the start request, the communication control space 6 registers to a process name table 15 the name of the server process 7_i received through the initiation report and the identifier of the space of the server process 7_i.

When a request to establish a logical communication path is received from the IWS 1_1, the communication control space 6 effects a retrieval on the process name table 15 by use of the server process name in the information transmitted together with the request and obtains the space of the server process 7_i. In the server process 7_i, a task called the communication control section 6a is beforehand initiated at the initiation report of the server process 7_i and is waiting for a report from the communication control space 6. The communication control space 6 transmits the report to the communication control section 6a in the server process 7_i by means of an interspace communication function (cross-memory post). On receiving the report from the communication control space 6, the communication control section 6a receives from the communication control space 6 the data indicating the request to establish a logical communication path, the data being transferred via an area of the virtual storage to be commonly used by the respective spaces 6 and 7_i. In the case where the request to open the logical communication path is to be accepted, an acknowledge answer is returned to the communication control space 6. The acknowledge answer is reported by use of a cross-memory post generated in a buffer of the common virtual storage. The communication control space 6 passes the acknowledge answer via the communication line 3_1 to the communication control section 5 of the IWS 1_1. The communication control section 5 notifies the process 4_i from which the request has been issued that the logical communication path has been established.

Through the procedures above, the process 4_i in the IWS 1_1 establishes the logical communication path to the server 7_i in the objective host 2_1. In this operation, the identification number of the logical communication path is transferred to the process 4_i in the IWS 1_1 and the server 7_i in the host 2_1.

When the logical communication path is established, the process 4_i in the IWS 1_1 is enabled to freely communicate with the server 7_i, and then like in the TSS, the server 7_i in the host 2_1 executes a processing for the function requested to the host 2_1. To effect a particular request for the server 7_i, the process 4_i generates request information in the virtual storage of the IWS 1_1, specifies the identification number of the logical communication path, and notifies the information to the communication control section 5. The communication control section 5 then transmits the information to the communication control space 6 in the host 2_1 via the communication line 3_1. Based on the identifier of the logical communication path, the information is recognized to be destined to the server 7_i; consequently, the information is delivered to the server

7_i.

The server 7_i is a server for handling a particular operation objective, for example, a mail server for the electronic mail or the multi-media document server for the multi-media document.

Since the server handles only the particular operation objective, it is possible to beforehand effect the OPEN processing by allocating the necessary files to the space of the server 7_i. To this end, in the space of each server 7_i, the fixed files required to handle the operation objectives of the server are subjected to the OPEN processing at the initiation. Fig. 7 shows the general configuration of the server 7_i, which comprises a communication accept section 6a for receiving data from the communication space 6, a server control task 7a for conducting analysis of the content of data transferred, and a plurality of request processing tasks 7b for implementing the functions of the requests.

The server task 7a issues a connect request to the communication control space 6 at the initiation, and as a result of the request, the communication accept section 6a is generated as a task. The server control task 7a then allocates the necessary files, opens the files, and waits for a request sent from the IWS side 1.

On receiving a processing request after the logical communication path to for example, the IWS 1_1 is opened, the server 7_i analyzes the request by use of the server control task 7a. As a result, if the pertinent request processing task 7b has already been assigned, the server 7_i passes the request to the request processing task 7b; otherwise, the server 7_i selects a free task 7b and assignes the request to the task 7b. The request processing task 7b effects the processing in accordance with the content of the request from the IWS 1_1. The processing result is returned to the process 4_i in the IWS 1_1 by means of the same logical communication path 3_1.

As described above, the server 7_i is a processing space in which the environment is arranged to meet the particular processing request (i.e. the file allocation and the OPEN processing of the files have been completed). Although each server 7_i can effect a processing execution only for a particular processing request, a plurality of requests can be received, namely, the requests from many IWS's can be simultaneously processed with respect to the space by use of, for example, the technology of the time slicing.

When the user's request to operate the IWS 1_1 is changed and a change-over function to change over control to another server becomes necessary, the analyze/status control section 11 in the IWS 1_1 initiates a process corresponding to a new Icon and then a connection to a new server in the host 2_1 is established in accordance with the IWS-host connect procedure. This connection is independent of the previous connection, namely, the previous connection may be retained to be used again when necessary. If such a connection is unnecessary, the connection is released as soon as possible. If necessary, a plurality of requests may be concurrently issued by use of two connections at the same time so as to be processed concurrently.

An example to effect the change-over of servers 7 in the IWS 1 is as follows.

An electronic mail is received (by the electronic mail server), the electronic mail is stored in an electronic cabinet (multi-media document store file) by means of the multi-media document server, and the document is printed by use of the print server.

This sequence of processing is implemented as follows. In accordance with a request issued from the IWS 1 operator, the electronic server is connected to the IWS 1, a request is issued to the electronic mail server to achieve the "electronic mail receive" function. Next, the multi-media server is connected and a request is made to the multi-media server to conduct the "store" function, and finally, the print server is connected and then a request is issued thereto to effect the "print" function.

According to the embodiment, since the user's request is analyzed in the IWS 1 and the processing environments (allocation and OPEN processing of files, development of tables in the memory, etc.) need not be set in each server of the host, and the server processes requests from a plurality of IWS's, which enables the number of server spaces to be limited, the load on the host can be reduced. Consequently, the number of IWS's to be connected to the host can be increased by the magnitude on the order of ten as compared with that of the prior art TSS.

Moreover, in some cases, the server satisfying the processing request need not be necessarily allocated to a particular host, namely, data such as documents and tables of each host can be freely and in the same procedure accessed through the server of the common kind disposed in each host.

Comparing now the method of the present invention with the prior art examples of Figs. 8 and 9, in the distributed interactive system of the present invention of Fig. 10, IWS's 1 are located on a plurality of terminal sides like in the on-line system to effect on the terminal sides the analysis of the request input from the terminal user, and on the host side, there are provided a communication program for conducting communications with the programs (processes) on the IWS 1 side and various servers each having a single function; consequently, processing requested from the plural IWS's can be collectively or concurrently be accomplished. As

a result, although the regular job and the irregular job can be processed in the present system, the range of the job may be limited depending on the kinds of servers.

It will be noted that the following advantages are obtained according to the present invention.

(1) In the conventional TSS, each TSS terminal has an individual space, and hence a great number of spaces are required. As a consequence, all spaces cannot be located in the memory and the processing is therefore executed by effecting a swap-out operation. Moreover, the change-over between spaces must be quite frequently achieved, which leads to a great overhead.

According to the present method, the number of server spaces is limited and hence the overhead associated with the number of spaces is reduced.

(2) Since the operation objective of each server is clearly specified, the environments necessary to implement the particular function such as the tables and files in the memory can be kept prepared in advance.

(3) Since the request is analyzed on the IWS side and the destination server is also determined therein, the special space to effect the request analysis is not necessary for the server on the host side, namely, the individual space required in the case of the TSS becomes unnecessary.

As a result, the host can dedicatedly effect the server processing.

Claims

1. A method of distributed interactive processing in a composite computer system (figure 1) which includes at least one work station (1), at least one host computer (2) supporting a group of server processes (7_1-7_ℓ), communication lines (3_1-3_n) connected between said at least one work station and said at least one host computer, and control means (5, 6) included in each said work station and each said host computer for controlling said communication lines to allow communications between said at least one work station and said at least one host computer,
comprising the steps of:
   - analysing in a said work station a user request entered at said work station to perform a specific processing and initiating a process (4_i) corresponding to said request in said work station (figure 4, 42-47, figure 5, 51-53), said user request not explicitly indicating whether said specific processing requires the execution of a said server process on a said host computer;
   - using data stored in advance (13) to identify the location of a processing resource required by said specific processing (figure 6, 61-63);
   and
   - where said processing resource is located in said work station, completing said specific processing within said work station (figure 6, 64-65)
   and
   - where said processing resource is located in a said host computer, sending process request information to said host computer (figure 6, 66-67), said information including a server process name;
   - said host computer responding to said process request information from said work station by carrying out the requested processing in a server process corresponding to said server process name and returning a result of said processing to said process within said work station.

2. A method according to claim 1, wherein
each said work station (1_1, 1_2, ..., 1_n) includes a function to execute a server process (4_i) of a relatively low level
and
each said host computer (2_1, 2_2, ..., 2_m) comprises a function to execute a server process (7_i) of a relatively high level.

3. A method according to claim 1, wherein
each said work station (1_1, 1_2, ..., 1_n) includes a step responsive to an input of a processing request to analyze the processing request and determine an existence location where a program corresponding to the processing request exists (Fig. 5).

4. A method according to claim 3, wherein
said request analyzing step (Fig. 5) includes a substep to be executed when it is detected in said work station (1_1, 1_2, ..., 1_n) a condition that a particular server process (7_1, 7_2, ..., 7_ℓ) of a particular one of said host computers (2_1, 2_2, ..., 2_m) is necessary for supplying said control means (5, 6) with a name of a processing objective process
and
said control means (5, 6) connects said particular server process (7_1, 7_2, ..., 7_ℓ) of said host computer (2_1, 2_2, ..., 2_m) to said work station (1_1, 1_2, ..., 1_n) in accordance with the name of the reception processing objective process, thereby allowing the processing re-

quest issued from a process of said work station (1_1, 1_2, ..., 1_n) to be received by said server process (7_1, 7_2, ..., 7_ℓ).

5. A method according to claim 4, wherein said server process (7_1, 7_2, ..., 7_ℓ) includes processes (7b) each prepared to process a particular kind of request.

6. A method according to claim 3, wherein said request analyzing step (Fig. 5) includes the substeps of

- obtaining a process name with reference to a table (12) including process names of processes (7_1, 7_2, ..., 7_ℓ) each implementing a request (61) to be inputted;
- attaining a name of a processing objective by executing the obtained process (62)
  and
- determining an existence location of the processing objective with reference to a directory (13, 63) provided in said at least one work station (1_1, 1_2, ..., 1_n).

7. A composite computer system for distributed interactive processing in performing a method according to one of the claims 1 to 6, including at least one work station (1_i), at least one host computer (2_i) supporting a group of server processes (7_1-7_ℓ), communication lines (3_1-3_n) connected between said at least one work station (1) and said at least one host computer (2), and control means (5, 6) controlling said communication lines (3_1, 3_n) to allow communications between said at least one work station (1_i) and said at least one host computer (2_i),
wherein each work station (1_i) comprises:

- input means (8-10) for receiving processing requests from a user;
- a process name table (12) in which names of processes to respectively implement processings for various processing requests are registered;
- means (11) responsive to said processing requests to determine a corresponding process name in said process name table (12) and to initiate a process having said process name;
- a directory (13) in which existence locations of the processing objectives corresponding to names of processing objectives including a file name and a program name are registered beforehand
- analysing means (4_1, 4_2, ..., 4_n) to in a said work station (1_i) analyse a user request entered at said work station (1_i) to perform a specific processing and initiating a process corresponding to said request in said work station (1_i), said user request not explicitly indicating whether said specific processing requires the execution of a said server process (figure 6, 61-63) on a said host computer (2_i) using data stored in advance (13) to identify the location of a processing resource required by said specific processing (figure 6, 64-65);
  and
- a processing resource located in said work station (1_i) for completing said specific processing within said work station
  and
- a processing resource located in said work station (1_i) for sending process request information to said host computer (2_i), said information including a server process name;

said host computer (2_i) responding to said process request information from said work station (1_i) by carrying out the requested processing in a server process corresponding to said server process name and returning a result of said processing to said process in said work station (1_i).

8. A system according to claim 7, wherein said host computer (2_i) comprises

- communication control means (6, 6a) for controlling the receiving and transferring of communication information from each said work station (1_i), comprising
  - a buffer existing in a virtual storage for storing information from said communication lines (3_i), said store information including a process name of a source issuing a request, a name of a server process as a communication destination and information indicating the content of the request
    and
  - server execution means (7_1, 7_2, ..., 7_ℓ) responsive to indication information from said communication control means (6, 6a) for executing the server process representing a process to implement a function for a particular operation objective.

9. A system according to claim 8, wherein said host computer (2_i) includes a name table (15) in which names of server processes and identifiers of spaces associated with the respective server processes are registered,
said server process execution means (7_1, 7_2,

..., 7_l) include a communication accept section (6a) for receiving information of a corresponding server spacewhich is obtained by said host computer communication control means (6, 6a) by referencing said process name table (15) in response to the information from said work station (1_i)
and
said communication accept section (6a) establishes, when an acknowledge answer is to be issued, a connection between the process in said work station (1_i) and the server in said host computer (2_i), thereby enabling the server to effect an accept processing of the processing request from the particular process.

Patentansprüche

1. Verfahren zur verteilten, interaktiven Verarbeitung in einem zusammengesetzten Computersystem (Fig. 1), das zumindest eine Arbeitsstation (1) aufweist, einen Verarbeitungscomputer (2), der eine Gruppe von Servervorgängen (7_1 - 7_l) unterstützt, Verbindungsleitungen (3_1 - 3_n), die zwischen die zumindest eine Arbeitsstation und den zumindest einen Verarbeitungscomputer geschaltet sind, und eine Steuerungseinrichtung (5, 6) in jeder Arbeitsstation und in jedem Verarbeitungscomputer zur Steuerung der Verbindungsleitungen, um die Verbindungen zwischen der zumindest einen Arbeitsstation und dem zumindest einen Verarbeitungscomputer zu ermöglichen, mit den Schritten:
   - Analysieren einer in die Arbeitsstation eingegebenen Benützeranfordnung in der Arbeitsstation, um eine spezifische Verarbeitung durchzuführen und um in der Arbeitsstation einen Vorgang (4_i) entsprechend der Anforderung zu beginnen (Fig. 4, 42-47, Fig. 5, 51-53), wobei die Benützeranforderung nicht ausdrücklich anzeigt, ob die spezifische Verarbeitung die Ausführung eines der Servervorgänge im Verarbeitungscomputer benötigt,
   - Verwendung vorweg gespeicherter Daten (13) zur Identifizierung des Orts einer Verarbeitungseinrichtung, die für die spezifische Verarbeitung benötigt wird (Fig. 6, 61-63);
     und
   - vollständiges Ausführen der spezifischen Verarbeitung innerhalb der Arbeitsstation (Fig. 6, 64-65), wenn sich die Verarbeitungseinrichtung in der Arbeitsstation befindet,
     und
   - Senden einer Vorgangsanforderungsinformation an den Verarbeitungscomputer (Fig. 6, 66-67), wenn sich die Verarbeitungseinrichtung im Verarbeitungscomputer befindet, wobei die Information den Namen eines Servervorgangs umfaßt;
   - wobei der Verarbeitungscomputer auf die von der Arbeitsstation kommende Vorgangsanforderungsinformation antwortet, indem entsprechend dem Namen des Servervorgangs die angeforderte Verarbeitung in einem Servervorgang ausgeführt wird und das Ergebnis der Verarbeitung an den Vorgang innerhalb der Arbeitsstation zurückgegeben wird.

2. Verfahren nach Anspruch 1, wobei
   jede der Arbeitsstationen (1_1, 1_2, ...1_n) eine Funktion zur Ausführung eines Servervorgangs (4_i) auf einem vergleichsweise niedrigem Niveau aufweist,
   und
   jeder der Verarbeitungscomputer (2_1, 2_2, ...2_m) eine Funktion zur Ausführung eines Servervorgangs (7_i) auf einem vergleichsweise hohen Niveau aufweist.

3. Verfahren nach Anspruch 1, wobei
   jede der Arbeitsstationen (1_1, 1_2, ...1_n) einen auf eine Eingabe einer Verarbeitungsanforderung ansprechenden Schritt zum Analysieren der Verarbeitungsanforderung und zur Bestimmung des Aufenthaltsortes, wo ein Programm entsprechend der Verarbeitungsanfrage existiert (Fig. 5), aufweist.

4. Verfahren nach Anspruch 3, wobei
   der Schritt des Analysierens der Anforderung (Fig. 5) einen Unterschritt aufweist, der dann auszuführen ist, wenn in der Arbeitsstation (1_1, 1_2, ...1_n) ein Zustand erfaßt wird, daß ein bestimmter Servervorgang (7_1, 7_2, ...7_l) eines bestimmten Verarbeitungscomputers (2_1, 2_2, ...2_m) notwendig ist, um der Steuerungseinrichtung (5, 6) einen Namen eines Verarbeitungszielvorgangs zuzuführen und
   die Steuerungseinrichtung (5, 6) den bestimmten Servervorgang (7_1, 7_2, ...7_l) des Verarbeitungscomputers (2_1, 2_2, ...2_m) entsprechend dem Namen des Empfangsverarbeitungszielvorgangs mit der Arbeitsstation (1_1, 1_2, ...1_n) verbindet, wodurch der Empfang der von einem Vorgang in der Arbeitsstation (1_1, 1_2, ...1_n) ausgegebenen Verarbeitungsanforderung durch den Servervorgang (7_1, 7_2, ...7_l) ermöglicht wird.

5. Verfahren nach Anspruch 4, wobei der Servervorgang (7_1, 7_2, ...7_l) Vorgänge (7b) umfaßt, die jeweils zur Verarbeitung einer bestimmten Art von Anforderung erzeugt wurden.

6. Verfahren nach Anspruch 3, wobei der Schritt des Analysierens einer Anforderung (Fig. 5) folgende Unterschritte enthält:
   - Ermitteln eines Vorgangsnamens bezugnehmend auf eine Tabelle 12, die Vorgangsnamen von Vorgängen (7_1, 7_2, ...7_l) enthält, von denen jeder eine einzugebende Anforderung (61) umsetzt;
   - Gewinnen eines Namens eines Verarbeitungsziels durch Ausführen des ermittelten Prozesses (62)
     und
   - Bestimmen eines Aufenthaltsorts des Verarbeitungsziels bezugnehmend auf ein Verzeichnis (13, 63), das in der zumindest einen Arbeitsstation (1_1, 1_2, ...1_n) vorgesehen ist.

7. Zusammengesetztes Computersystem für die verteilte, interaktive Verarbeitung mittels Durchführung eines Verfahrens nach einem der Ansprüche 1 bis 6,
   mit zumindest einer Arbeitsstation (1_i), zumindest einem Verarbeitungscomputer (2_i), der eine Gruppe von Servervorgängen (7_1 - 7_l) unterstützt, Verbindungsleitungen (3_1 - 3_n), die zwischen die zumindest eine Arbeitsstation (1) und den zumindest einen Verarbeitungscomputer (2) geschaltet sind, und einer Steuerungseinrichtung (5, 6), die die Verbindungsleitungen (3_1, 3_n) steuern, um die Verbindung zwischen der zumindest einen Arbeitsstation (1_i) und dem zumindest einen Verarbeitungscomputer (2_i) zu ermöglichen,
   wobei jede Arbeitsstation (1_i) aufweist:
   - eine Eingabeeinrichtung (8 - 10) zum Empfangen von Verarbeitungsanforderungen von einem Anwender;
   - eine Vorgangsnamentabelle (12), in der Namen von Vorgängen für die jeweilige Umsetzung von Verarbeitungen für die verschiedenen Verarbeitungsanforderungen gespeichert sind;
   - eine Einrichtung (11), die auf die Verarbeitungsanforderungen anspricht, um einen entsprechenden Vorgangsnamen in der Vorgangsnamentabelle (12) zu bestimmen und um einen Vorgang, der diesen Vorgangsnamen hat, zu beginnen,
   - ein Verzeichnis (13), in dem vorweg die Aufenthaltsorte von Verarbeitungszielen entsprechend den Namen der Verarbeitungsziele einschließlich eines Dateinamens und eines Programmnamens gespeichert werden,
   - eine Analysiereinrichtung (4_1, 4_2, ...4_n), um in der Arbeitsstation (1_i) eine in der Arbeitsstation (1_i) eingegebene Benutzeranforderung zu analysieren, um eine spezifische Verarbeitung durchzuführen und um einen Vorgang entsprechend der Anforderung in der Arbeitsstation (1_i) zu beginnen, wobei die Benutzeranforderung nicht explizit angibt, ob die spezifische Verarbeitung die Ausführung eines der Servervorgänge (Fig. 6, 61-63) im Verarbeitungscomputer (2_i) unter Verwendung der vorab gespeicherten Daten (13) zur Identifizierung des Orts einer durch die spezifische Verarbeitung (Fig. 6, 64-65) benötigten Verarbeitungseinrichtung benötigt;
     und
   - eine sich in der Arbeitsstation (1_i) befindende Verarbeitungseinrichtung zur Beendigung der spezifischen Verarbeitung innerhalb der Arbeitsstation und
   - eine sich in der Arbeitsstation (1_i) befindende Verarbeitungseinrichtung zum Senden einer Vorgangsanforderungsinformation an den Verarbeitungscomputer (2_i), wobei die Information einen Servervorgangsnamen umfaßt,
     wobei der Verarbeitungscomputer (2_i) auf die von der Arbeitsstation (1_i) kommende Prozeßanforderungsinformation reagiert, indem die angeforderte Verarbeitung in einem dem Servervorgangsnamen entsprechenden Servervorgang ausgeführt wird und das Ergebnis der Verarbeitung an den Vorgang in der Arbeitsstation (1_i) zurückgegeben wird.

8. System nach Anspruch 7, bei dem der Verarbeitungscomputer (2_i) aufweist
   - eine Verbindungssteuerungseinrichtung (6, 6a) zur Steuerung des Empfangs und des Übertragens von Verbindungsinformation von jeder der Arbeitsstationen (1_i), mit
     - einem in einem virtuellen Speicher existierenden Puffer zur Speicherung von von den Verbindungsleitungen (3_i) kommenden Informationen, wobei die gespeicherte Information den Vorgangsnamen einer eine Anforderung ausgebenden Quelle aufweist, den Namen eines Servervorgangs als Verbindungszielort sowie eine Information, die den Inhalt der Anforderung angibt, und

- einer Serverausführungseinrichtung (7_1, 7_2, ...7_l), die nach Maßgabe einer Angabeinformation aus der Verbindungssteuerungseinrichtung (6, 6a) arbeitet, zur Ausführung des Servervorgangs, der einen Vorgang zur Umsetzung einer Funktion für ein bestimmtes Vorgangsziel darstellt.

9. System nach Anspruch 8, wobei
der Verarbeitungscomputer (2_i) eine Namenstabelle (15) aufweist, in der Namen von Servervorgängen und Identifizierer von mit den jeweiligen Servervorgängen verknüpften Räumen gespeichert sind,
wobei die Servervorgangsausführungseinrichtung (7_1, 7_2, ...7_l) einen Verbindungsannahmeabschnitt (6a) aufweist zum Empfangen von Information eines entsprechenden Serverraums, die durch die Verbindungssteuerungseinrichtung (6, 6a) des Host-Computers durch Bezugnahme auf die Vorgangsnamentabelle (15) nach Maßgabe der Information aus der Arbeitsstation (1_i) ermittelt wird, und
wobei
der Verbindungsannahmeabschnitt (6a) eine Verbindung zwischen dem Vorgang in der Arbeitsstation (1_i) und dem Server im Verarbeitungscomputer (2_i) aufbaut, wenn eine Bestätigungsantwort auszugeben ist, wodurch der Server in die Lage versetzt wird, eine Annahmeverarbeitung der Verarbeitungsanforderung des besonderen Vorgangs durchzuführen.

Revendications

1. Procédé de traitement interactif réparti dans un système informatique composite (figure 1), qui comprend au moins un poste de travail (1), au moins un ordinateur central (2) supportant un groupe de processus de serveurs (7_1-7_l), des lignes de communication (3_1-3_n) branchées entre ledit au moins un poste de travail et ledit au moins un ordinateur central, et des moyens de commande (5, 6) contenus dans chacun desdits postes de travail et chacun desdits ordinateurs centraux pour commander lesdites lignes de communication de manière à permettre les communications entre ledit au moins un poste de travail et ledit au moins un ordinateur central, comprenant les étapes consistant à :

- analyser, dans l'un desdits postes de travail, une demande d'utilisateur introduite dans ledit poste de travail pour exécuter un traitement spécifique et déclencher un processus (4_i) correspondant à ladite demande dans ledit poste de travail (figure 4, 42-47, figure 5, 51-53), ladite demande d'utilisateur n'indiquant pas de façon explicite si ledit traitement spécifique requiert l'exécution de l'un desdits processus de serveurs dans l'un desdits ordinateurs centraux;
- utiliser des données mémorisées par avance (13) pour identifier l'emplacement d'une ressource de traitement qui est requise par ledit traitement spécifique (figure 6, 61-63); et
- lorsque ladite ressource de traitement est située dans ledit poste de travail, achever ledit traitement spécifique dans ledit poste de travail (figure 6, 64-65), et
- lorsque ladite ressource de traitement est située dans ledit ordinateur central, envoyer une information de demande de processus audit ordinateur central (figure 6, 66-67), ladite information contenant un nom de processus de serveur;
- ledit ordinateur central répondant à ladite information de demande de processus provenant dudit poste de traitement, par l'exécution du traitement demandé dans un processus de serveur correspondant audit nom de processus de serveur,et renvoyant un résultat dudit traitement audit processus à l'intérieur dudit poste de travail.

2. Procédé selon la revendication 1, selon lequel chacun desdits postes de travail (1_1, 1_2, ..., 1_n) inclut une fonction consistant à exécuter un processus de serveur (4_i) possédant un niveau relativement bas, et
chacun desdits ordinateurs centraux (2_1, 2_2, ..., 2_m) comprend une fonction servant à exécuter un processus de serveur (7_i) possédant un niveau relativement élevé.

3. Procédé selon la revendication 1, selon lequel chacun desdits postes de travail (1_1, 1_2, ..., 1_n) inclut une étape répondant à une entrée d une demande de traitement pour analyser la demande de traitement et déterminer un emplacement d'existence, où existe un programme correspondant à la demande de traitement (figure 5).

4. Procédé selon la revendication 3, selon lequel ladite étape d'analyse de demande (figure 5) inclut une étape secondaire devant être exécutée lorsque dans ledit poste de travail (1_1, 1_2, ..., 1_n) est détectée une condition selon laquelle un processus de serveur particulier (7_1, 7_2, ..., 7_l) de l'un particulier desdits ordinateurs centraux (2_1, 2_2, ..., 2_m) est nécessaire pour l'envoi d'un nom d'un processus objectif

de traitement auxdits moyens de commande (5, 6), et
lesdits moyens de commande (5, 6) raccordent ledit processus de serveur particulier (7_1, 7_2, ..., 7_n) dudit ordinateur central (2_1, 2_2, ..., 2_m) audit poste de travail (1_1, 1_2, ..., 1_n) en fonction du processus objectif de traitement de réception, ce qui permet la réception de la demande de traitement, provenant d'un processus dudit poste de travail (1_1, 1_2, ..., 1_n), par ledit processus de serveur (7_1, 7_2, ..., 7_n).

5. Procédé selon la revendication 4, selon lequel ledit processus de serveur (7_1, 7_2, ..., 7_n) inclut des processus (7b) préparés chacun pour traiter un type particulier de demande.

6. Procédé selon la revendication 3, selon lequel ladite étape d'analyse de demande (figure 5) inclut les étapes secondaires consistant à
- obtenir un nom de processus par référence à une table (12) contenant des noms de processus (7_1, 7_2, ..., 7_l) mettant en oeuvre chacun une demande (61) devant être introduite;
- obtenir un nom d'un objectif de traitement moyennant l'exécution du processus obtenu (62), et
- déterminer un emplacement d'existence de l'objectif de traitement en référence à un répertoire (13, 63) prévu dans ledit au moins un poste de travail (1_1, 1_2, ..., 1_n).

7. Système informatique composite pour un traitement interactif réparti, pour la mise en oeuvre d'un procédé selon l'une des revendications 1 à 6, comprenant au moins un poste de travail (1), au moins un ordinateur central (2_i) supportant un groupe de processus de serveurs (7_1-7_l), des lignes de communication (3_1-3_n) branchées entre ledit au moins un poste de travail (1) et ledit au moins un ordinateur central (2), et des moyens de commande (5, 6) commandant lesdites lignes de communication (3_1, 3_n) pour permettre l'établissement de communications entre ledit au moins un poste de travail (1_i) et ledit au moins un ordinateur central (2_i), et dans lequel chaque poste de travail (1_i) comprend :
- des moyens d'entrée (8-10) pour recevoir des demandes de traitement de la part d'un utilisateur;
- une table (12) de noms de processus, dans laquelle sont enregistrés des noms de processus servant à mettre en oeuvre respectivement des traitements pour différentes demandes de traitement;
- des moyens (11) répondant auxdites demandes de traitement pour déterminer un nom de processus correspondant dans ladite table (12) de noms de processus et déclencher un processus possédant ledit nom;
- un répertoire (13), dans lequel des emplacements d'existence des objectifs de traitement correspondant à des noms d'objectifs de traitement incluant un nom de fichier et un nom de programme, sont enregistrés par avance,
- des moyens d'analyse (4_1, 4_2, ..., 4_n) pour analyser, dans l'un desdits postes de travail (1_i), une demande d'utilisateur introduite dans ledit poste de travail (1_i) pour exécuter un traitement spécifique et déclencher un processus correspondant à ladite demande dans ledit poste de travail (1_i), ladite demande d'utilisateur n'indiquant pas explicitement si ledit traitement spécifique requiert l'exécution dudit processus de serveur (figure 6, 61-63) dans ledit ordinateur central (2_i) moyennant l'utilisation ce données, mémorisées par avance (13) pour identifier l'emplacement d'une ressource de traitement requise par ledit traitement spécifique (figure 6, 64-65); et
- une ressource de traitement située dans ledit poste de travail (1_i) pour achever ledit traitement spécifique dans ledit poste de travail, et
- une ressource de traitement située dans ledit poste de travail (1_i) pour envoyer une information de demande de processus audit ordinateur central (2_i), ladite information incluant un nom de processus de serveur;
- ledit ordinateur central (2_i) répondant à ladite information de demande de processus provenant dudit poste de travail (1_i) en exécutant le traitement demandé dans un processus de serveur correspondant audit nom de processus de serveur, et renvoyant le résultat dudit traitement audit processus dans ledit poste de travail (1_i).

8. Système selon la revendication 7, dans lequel ledit ordinateur central (2_i) comprend
- des moyens de commande de communication (6, 6a) pour commander la réception et le transfert d'une information de communication à partir de chacun desdits postes de travail (1_i), comprenant
  - un tampon existant dans une mémoire virtuelle pour mémoriser une informa-

tion provenant desdites lignes de communication (3_i), ladite information de mémorisation comprenant un nom de processus d'une source délivrant une demande, un nom d'un processus ce serveur en tant que destination de communication et une information indiquant le contenu de la demande, et

- des moyens d'exécution de serveur (7_1, 7_2, ..., 7_n) aptes à répondre à une information d'indication délivrée par lesdits moyens de commande de communication (6, 6a) pour l'exécution du processus de serveur représentant un processus pour la mise en oeuvre d'une fonction pour un objectif d'opération particulière.

9. Système selon la revendication 8, dans lequel ledit ordinateur central (2_i) comprend une table de noms (15),dans laquelle des noms de processus de serveurs et des identifiants d'espaces associés aux processus respectifs de serveurs sont enregistrés,
lesdits moyens d'exécution de processus de serveurs (7_1, 7_2, ..., 7_n) incluent une section d'acceptation de communication (6a) pour la réception d'une information d'un espace de serveur correspondant, qui est obtenue par lesdits moyens (6, 6a) de commande de communication dudit ordinateur central, par référence à ladite table (15) de noms de processus en réponse à l'information délivrée arrivant par ledit poste de travail (1_i), et
ladite section d'acceptation de communication (6a) établit, lorsqu'une réponse d'accusé de réception doit être délivrée, une connexion entre le processus dans ledit poste de travail (1_i) et le serveur dans ledit ordinateur central (2_i), ce qui permet au serveur d'exécuter un traitement d'acceptation de la demande de traitement à partir du processus particulier.

FIG. 1

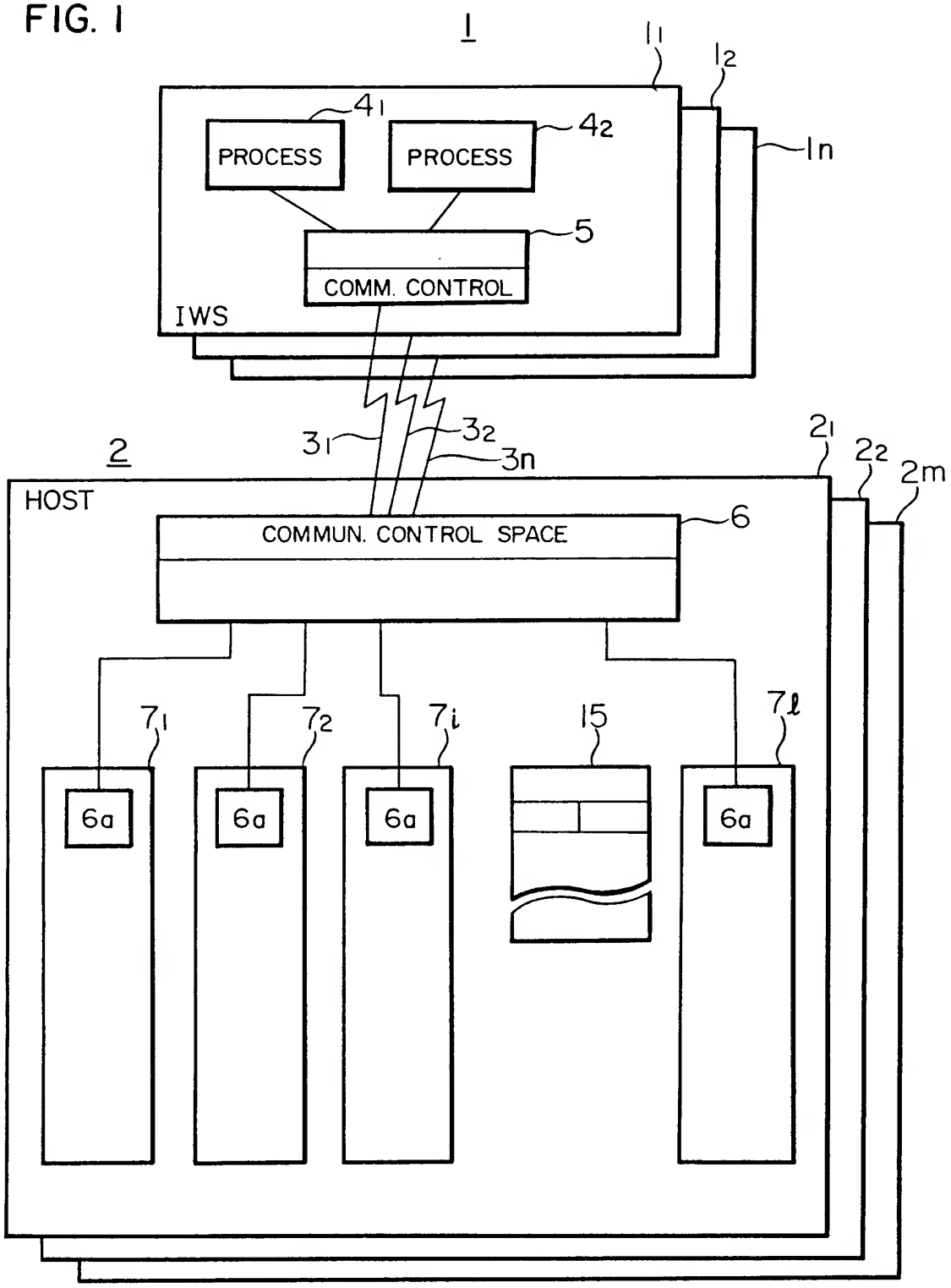

1
11
12
41
42
PROCESS
PROCESS
1n
5
COMM. CONTROL
IWS
31
32
3n
2
21
22
2m
HOST
COMMUN. CONTROL SPACE
6
71
72
7i
15
7l
6a
6a
6a
6a

FIG. 2

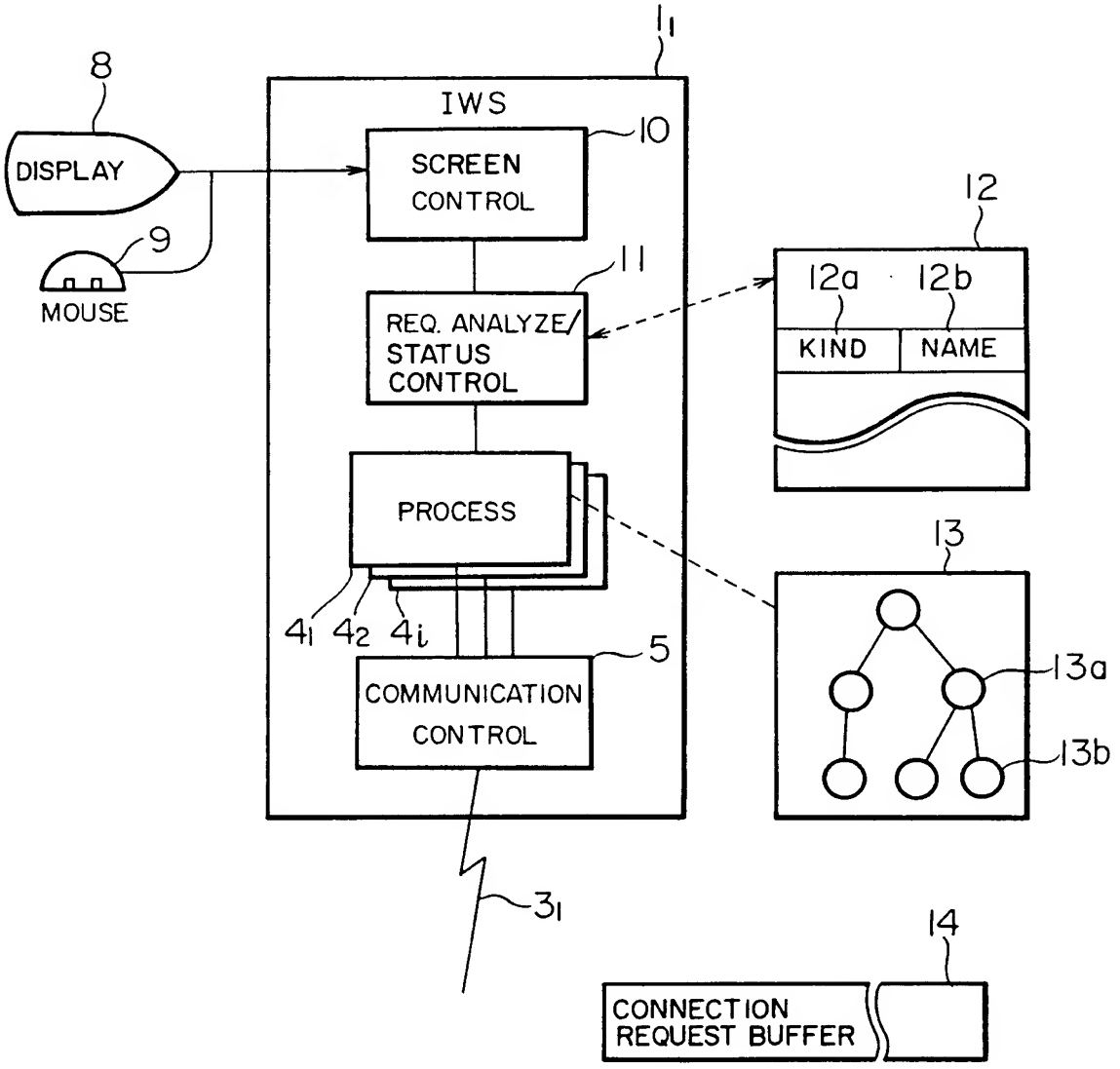

1₁
8
IWS
DISPLAY
10
SCREEN CONTROL
12
9
11
12a
12b
MOUSE
REQ. ANALYZE/ STATUS CONTROL
KIND
NAME
PROCESS
13
4₁ 4₂ 4ᵢ
5
13a
COMMUNICATION CONTROL
13b
3₁
14
CONNECTION REQUEST BUFFER

FIG. 3

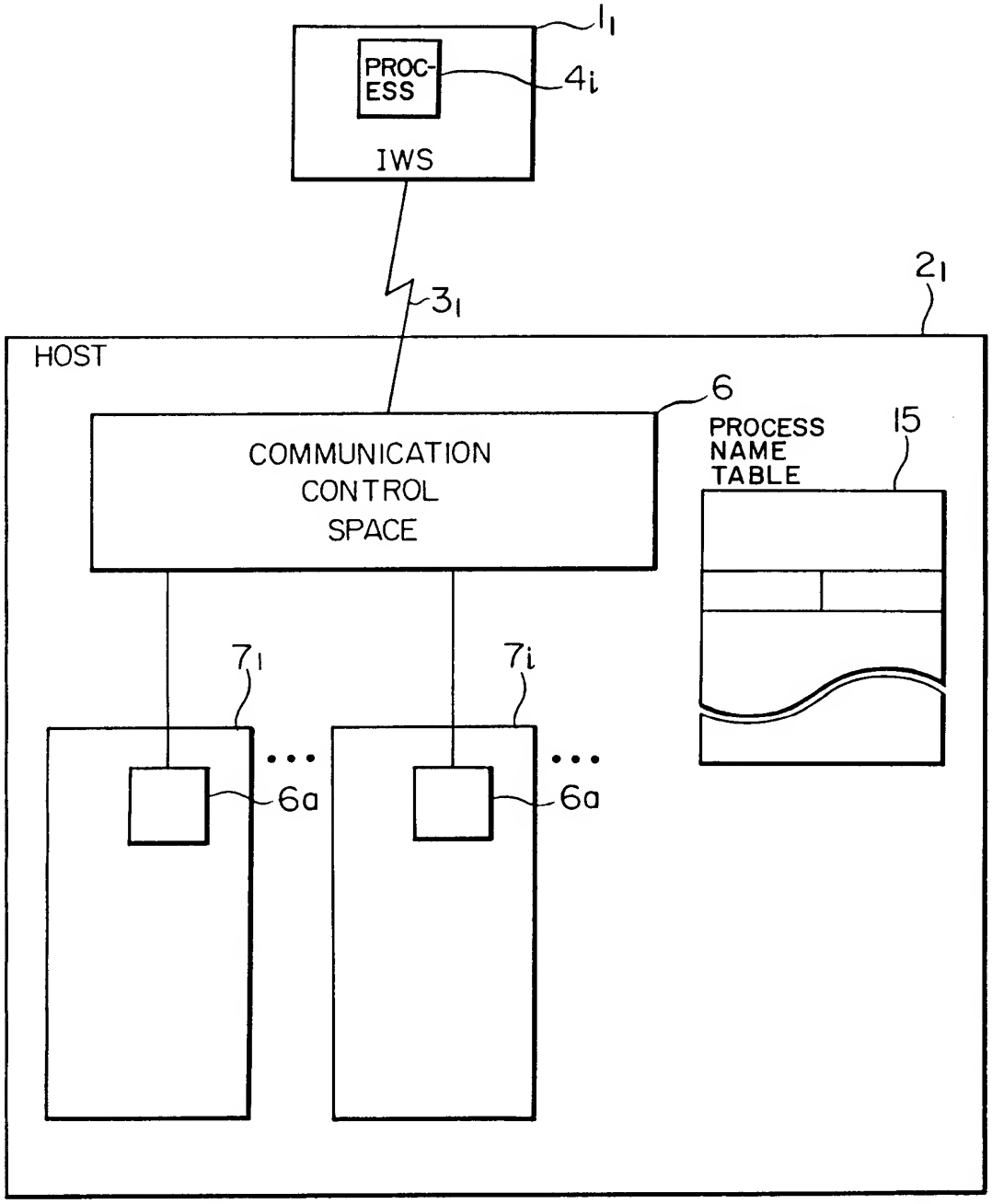

1i
PROC-
ESS
4i
IWS
3i
2i
HOST
6
COMMUNICATION
CONTROL
SPACE
PROCESS
NAME
TABLE
15
7i
7i
6a
6a

FIG. 4

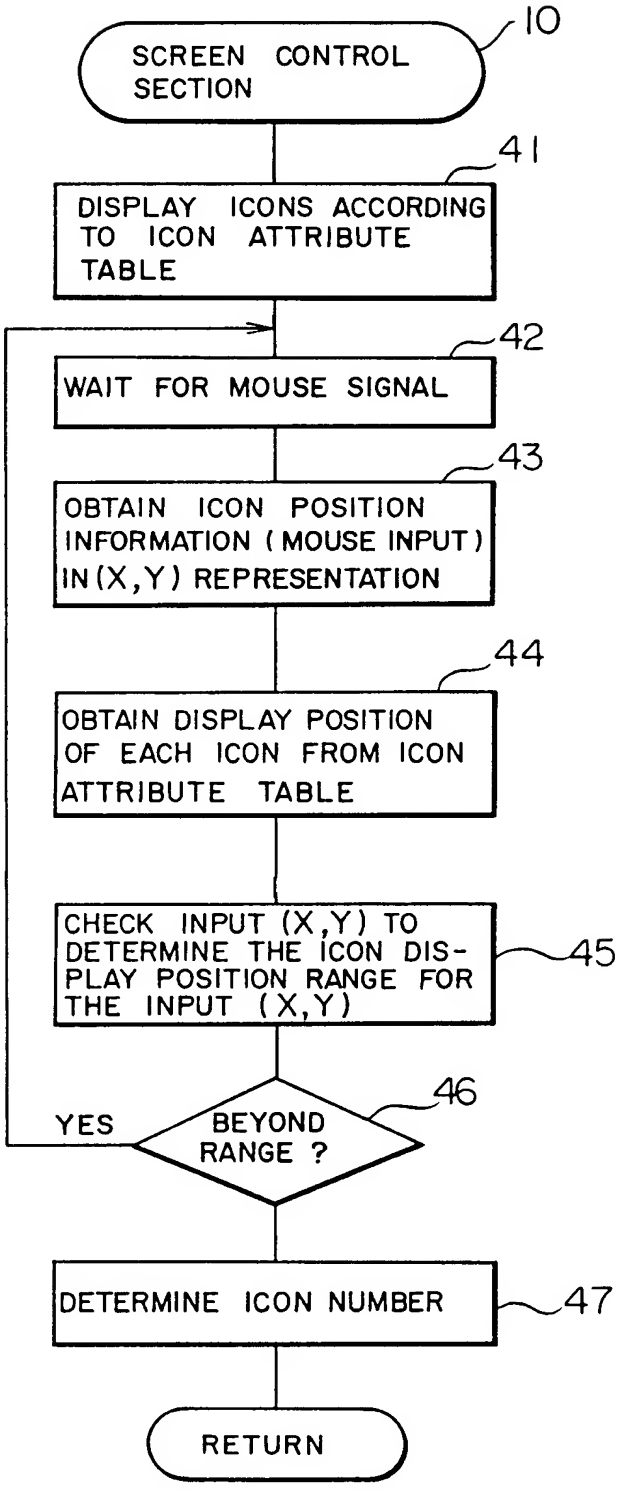

SCREEN CONTROL SECTION
10
DISPLAY ICONS ACCORDING TO ICON ATTRIBUTE TABLE
41
WAIT FOR MOUSE SIGNAL
42
OBTAIN ICON POSITION INFORMATION (MOUSE INPUT) IN (X,Y) REPRESENTATION
43
OBTAIN DISPLAY POSITION OF EACH ICON FROM ICON ATTRIBUTE TABLE
44
CHECK INPUT (X,Y) TO DETERMINE THE ICON DISPLAY POSITION RANGE FOR THE INPUT (X,Y)
45
YES
BEYOND RANGE ?
46
DETERMINE ICON NUMBER
47
RETURN

FIG. 5

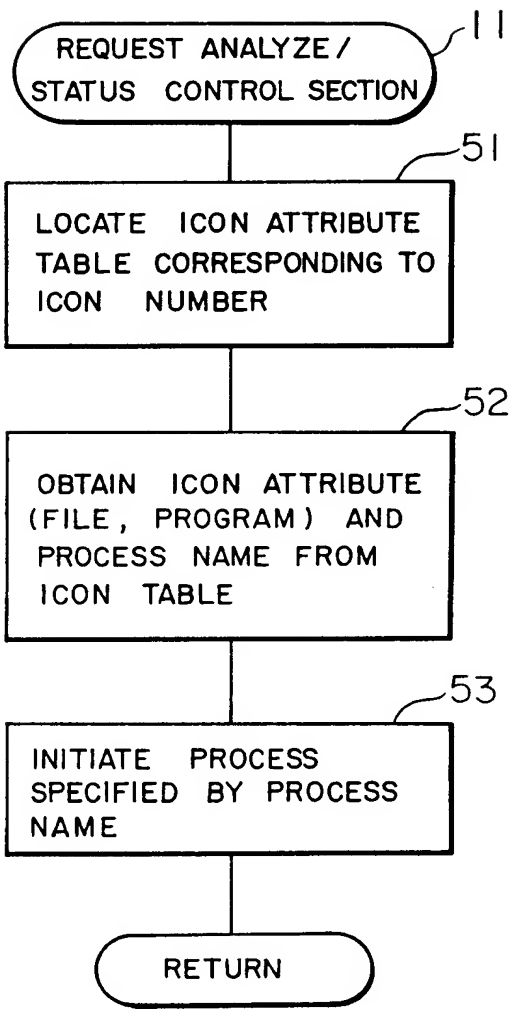

REQUEST ANALYZE / STATUS CONTROL SECTION
11
LOCATE ICON ATTRIBUTE TABLE CORRESPONDING TO ICON NUMBER
51
OBTAIN ICON ATTRIBUTE (FILE, PROGRAM) AND PROCESS NAME FROM ICON TABLE
52
INITIATE PROCESS SPECIFIED BY PROCESS NAME
53
RETURN

FIG. 6

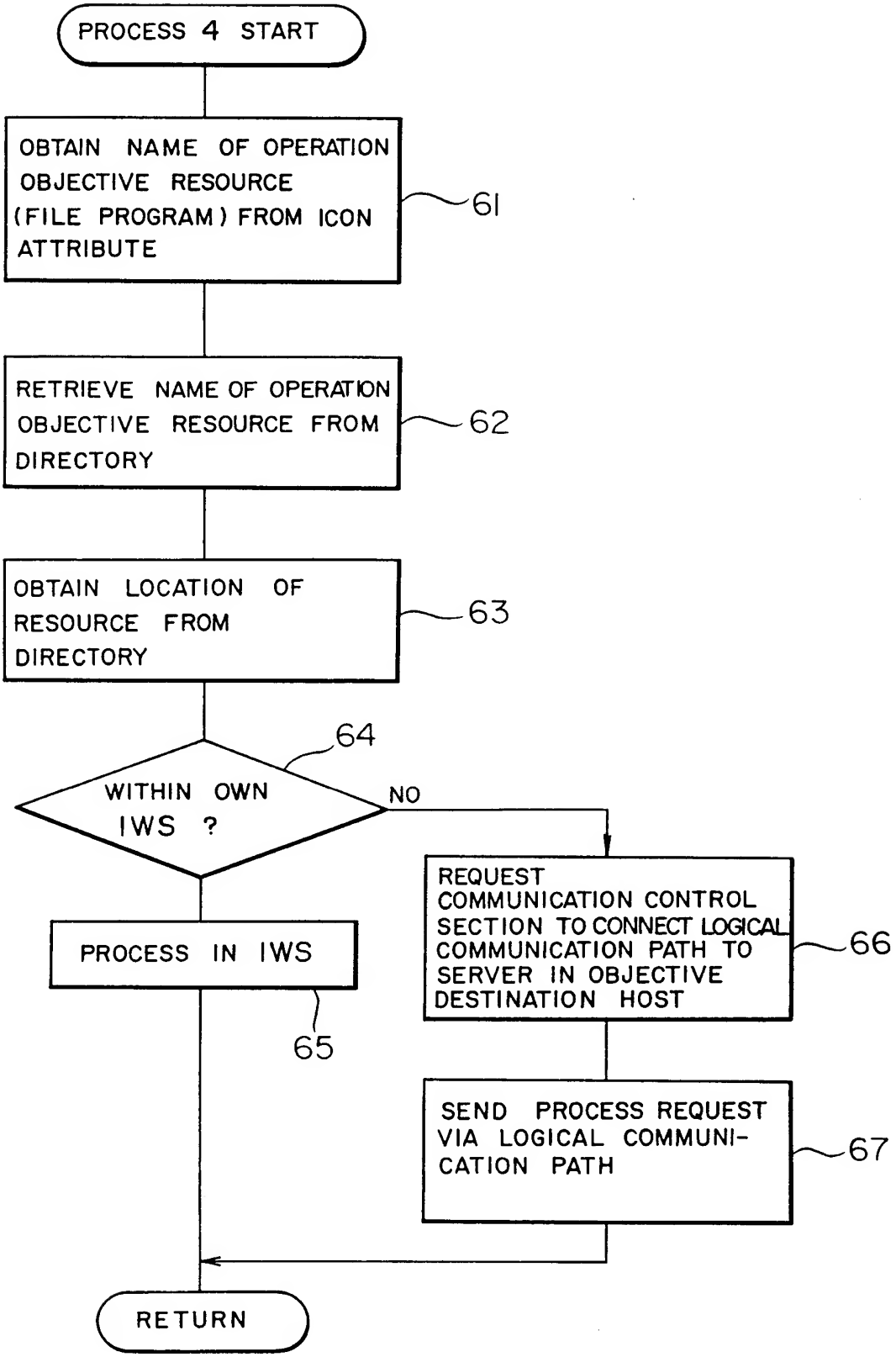

PROCESS 4 START
OBTAIN NAME OF OPERATION OBJECTIVE RESOURCE (FILE PROGRAM) FROM ICON ATTRIBUTE
61
RETRIEVE NAME OF OPERATION OBJECTIVE RESOURCE FROM DIRECTORY
62
OBTAIN LOCATION OF RESOURCE FROM DIRECTORY
63
64
WITHIN OWN IWS ?
NO
PROCESS IN IWS
65
REQUEST COMMUNICATION CONTROL SECTION TO CONNECT LOGICAL COMMUNICATION PATH TO SERVER IN OBJECTIVE DESTINATION HOST
66
SEND PROCESS REQUEST VIA LOGICAL COMMUNICATION PATH
67
RETURN

FIG. 7

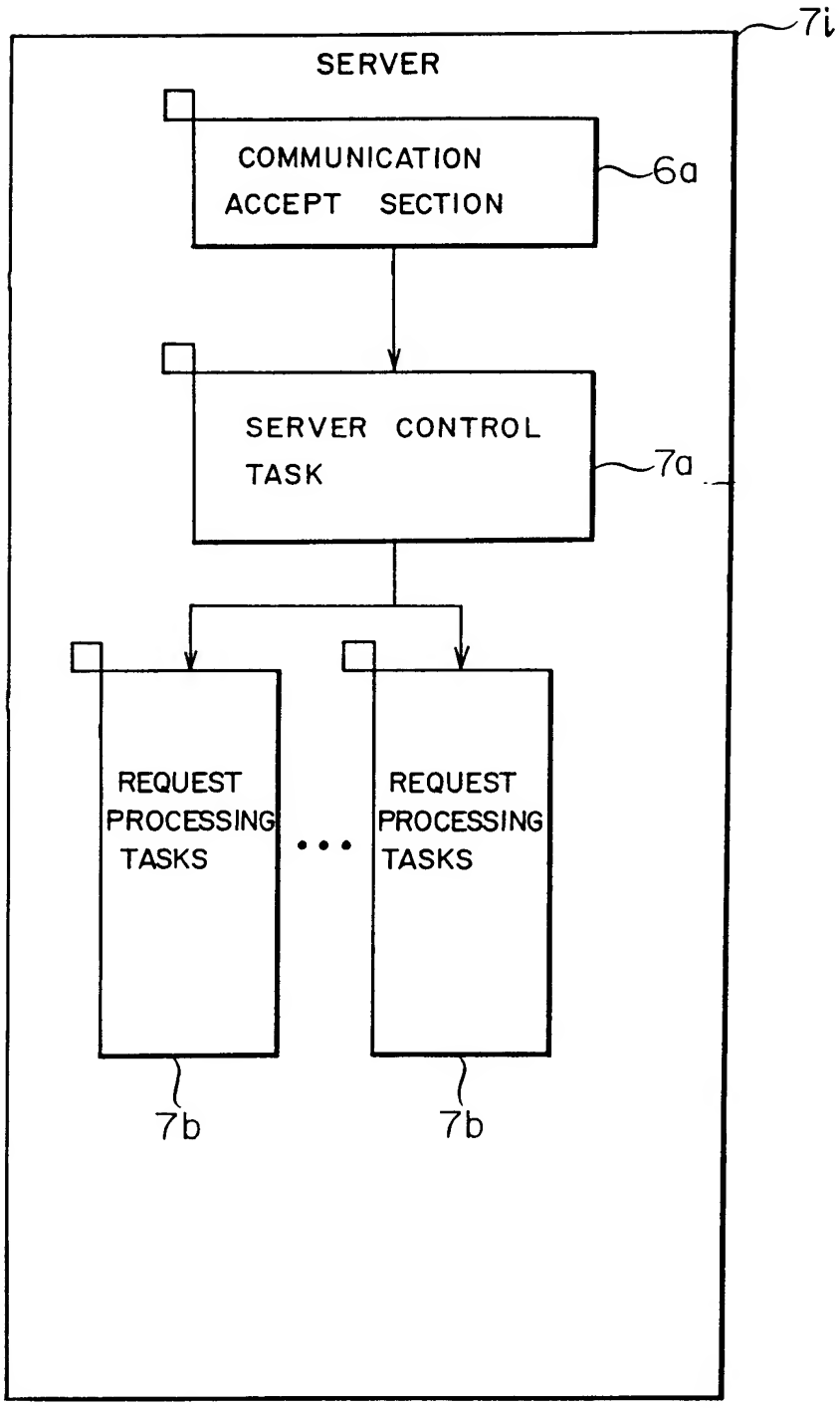

7i
SERVER
COMMUNICATION ACCEPT SECTION
6a
SERVER CONTROL TASK
7a
REQUEST PROCESSING TASKS
REQUEST PROCESSING TASKS
7b
7b

FIG. 8
PRIOR ART

ON-LINE TRANSACTION PROCESSING

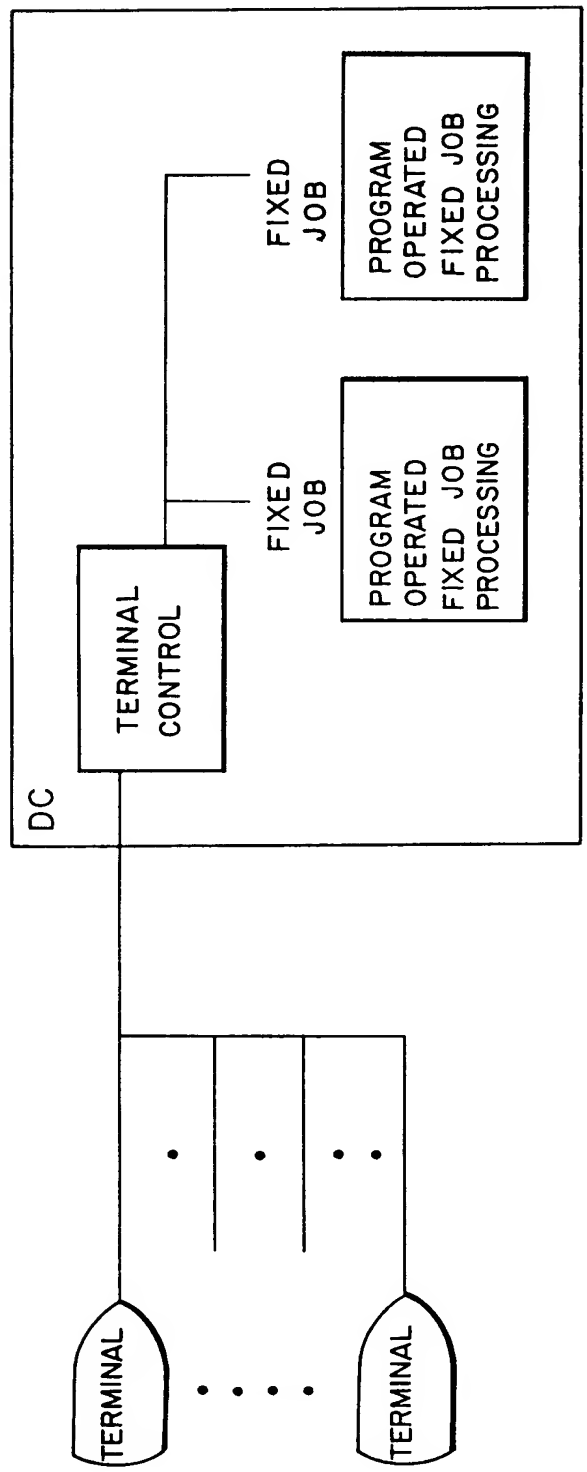


FIG. 9
PRIOR ART

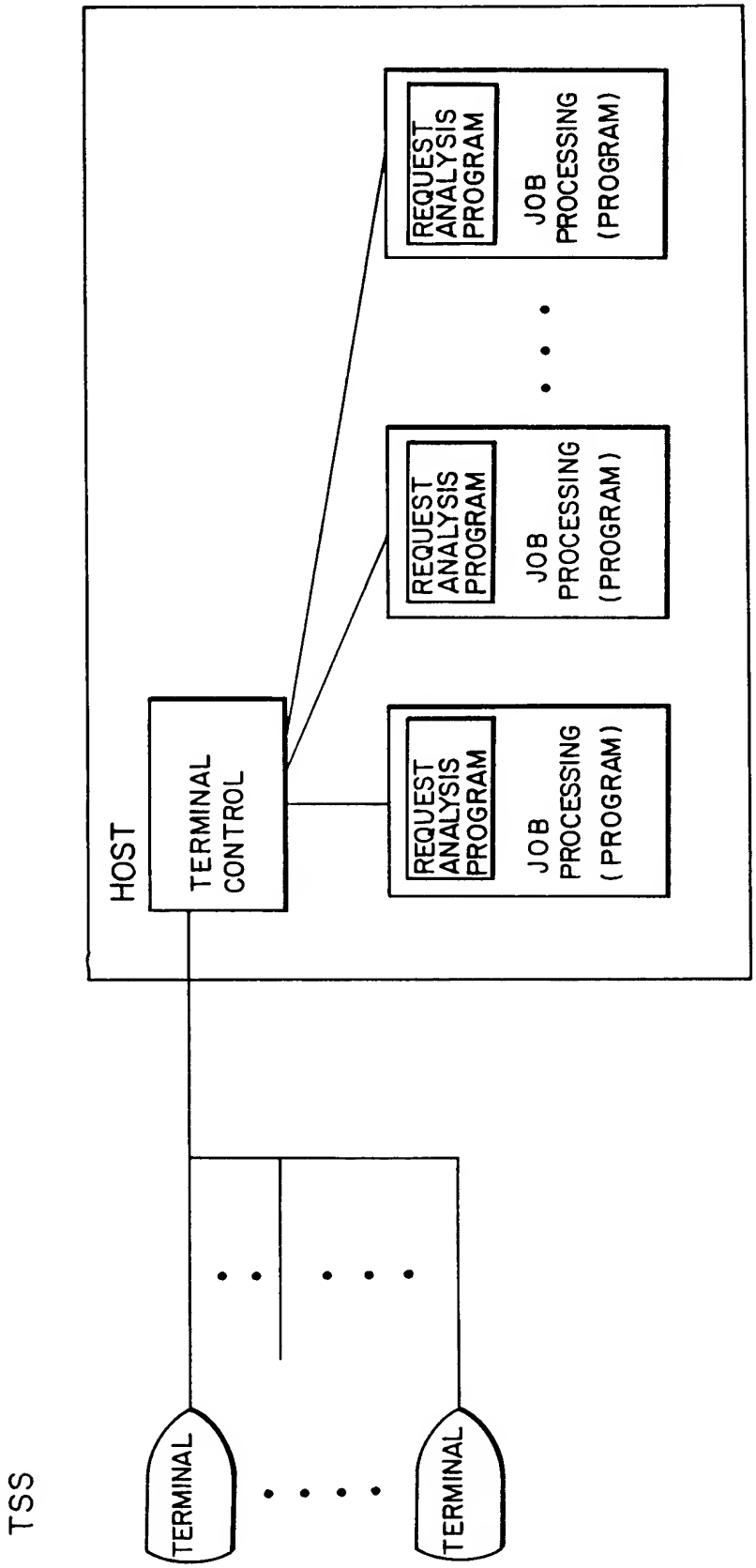
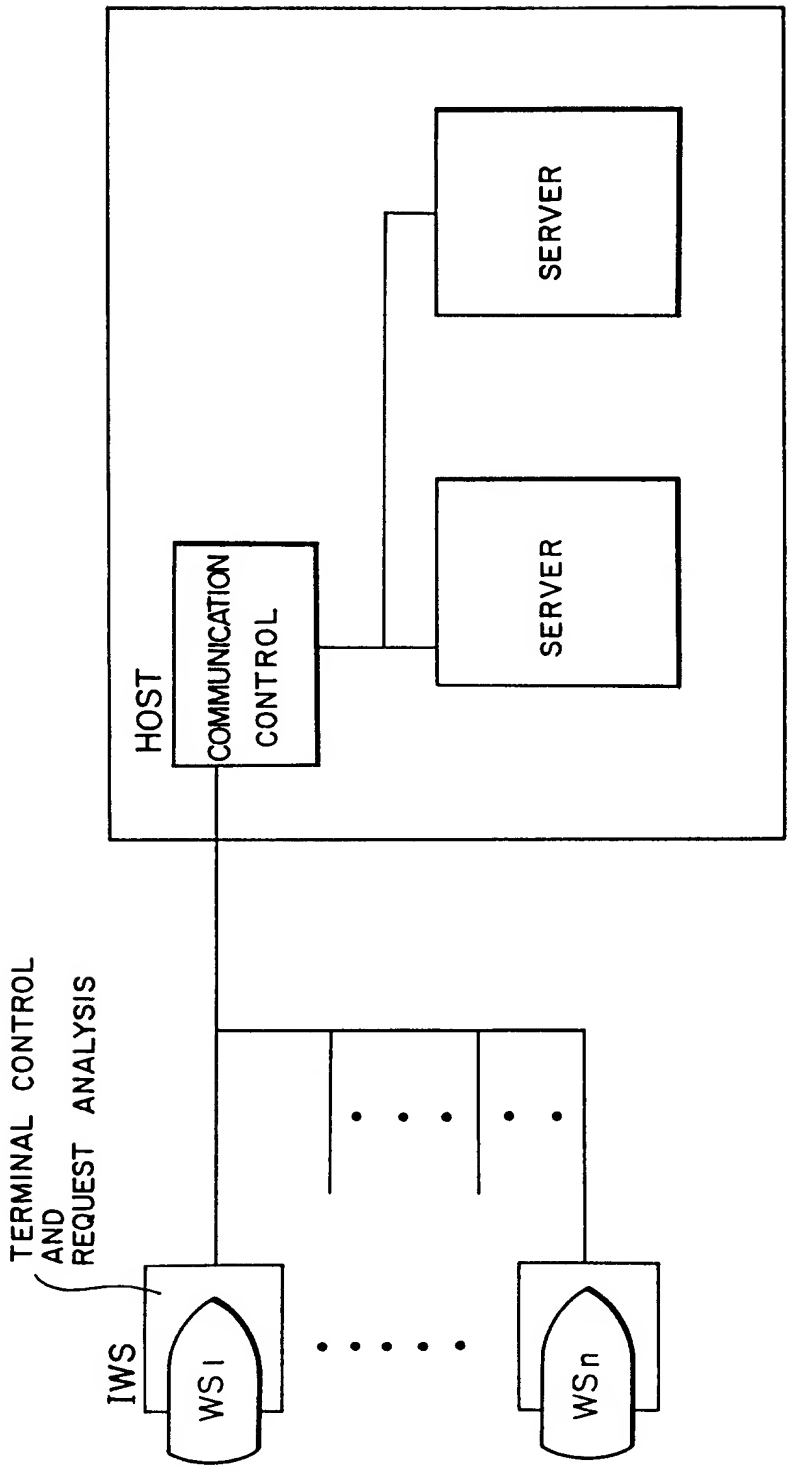


FIG. 10


HOST
COMMUNICATION CONTROL
SERVER
SERVER
TERMINAL CONTROL AND REQUEST ANALYSIS
IWS
WS 1
WSn